

DM 10112106
101032, 881

Page 4

12
Please amend the paragraph that starts at page 9, line 6 as follows:

In this specification and in the appended claims, the term entry refers to a piece of information relating to one set of data packets. An entry typically comprises

5 information at least about data packet headers. The term "set of data packets" ~~set of data packets~~ on the other hand refers to data packets, which are related to each other, such as data packets of a packet data connection, data packets of a communication session comprising a plurality of packet data connections, data packets of a plurality of packet data connections of a secure tunnel, or any other

10 suitable set of data packets. The terms "state data structure", "node-specific data structure" or "common data structure" ~~term state data structure, node specific data structure or common data structure~~ refer to a data structure, whose entries represent sets of data packets handled in a network element. Such data structures may be, for example, a table or a linked list or any other more versatile data

15 structure.

Please amend the paragraph that starts at page 12, line 11 as follows:

In step 212 in a plurality of entries of said node-specific and common data structures is maintained distribution information relating to the distribution identifier, which corresponds to the set of data packets related to the respective entry. This step ~~makes it enables that it is~~ possible to identify to which node a particular entry in the node-specific and common data structures may belong ~~belongs~~ at a given moment in time. This is a key element for making it possible to implement the invention. Since there is knowledge of the grounds of distributing

20 sets of data packets to different nodes, it is possible to keep track ~~on~~ which entries belong to which node even if distribution of the sets of data packets varies dynamically.

25

Please amend the paragraph that starts at page 13, line 13 as follows:

30

Figure 4 illustrates as an example a flowchart of a method adding entries to the state data structures. In general, a new entry representing state information related to a set of data packets is added to a state data structure as a first data packet of the set of data packets is handled. The state data structure entry is then used for handling the other data packets of the set of data packets. After handling all data packets of the set, ~~set are~~ the corresponding entry is cleared from the state data structure. It is well known how to determine which data packets cause adding a new entry and which are handled according to the state data structure. For

35

example, a data packet may be compared against entries in a state data structure and if a corresponding entry is not found a new entry needs to be made. If a corresponding entry is found the data packet is handled accordingly. Additionally, it is possible to perform a further check whether the new entry would be allowable
5 e.g. on the basis of rules, before adding the new entry in the node-specific data structure. Furthermore, some data packets may require special handling and/or special entries in the state data structure. For example a specific separate code portion may be required for handling some set of data packets and such code portion may create to the state data structure new entries for some other set of data
10 packets on the basis of data packets it is handling. Creating an entry for FTP data connection on the basis of related FTP control connection is an example of such handling of data packets. Such handling of data packets is however beyond the scope of this invention and is not addressed here any further. Considering the invention it is irrelevant on what basis an entry is added or cleared in a state data
15 structure, since the invention does not concern determining the entries, but rather handling existing entries.

Please amend the paragraph that starts at page 16, line 35 as follows:

20 The nodes Node1, Node2 and Node3 are provided with the ability to possibility communicate information between the common data structures of other nodes and between their own common and node-specific data structures. The common data structures change dynamically in time responsive to adding new entries or clearing old entries in the node-specific data structures due to handling sets of data packets as was explained in more detail in connection with Figure 4. Whereas node-
25 specific data structures change dynamically in time due to handling sets of data packets, and also due to reallocation of distribution identifiers as is explained in more detail in connection with Figure 6. The state data structures of the nodes are presented as examples only and it should be clear that it is possible to implement the state data structures also in some other way within the scope of the invention.
30

Please amend the paragraph that starts at page 16, line 36 as follows:

In order to clarify the operation during reallocation of distribution identifiers let's consider a situation where distribution identifiers 1, 2 and 3 are initially allocated
35 to a node 1 and distribution identifiers 4 and 5 to a node 2. Due to load balancing the distribution identifier 3 needs to be reallocated to the node 2, and consequently the sets of data packets relating to the distribution identifier 3 need to be

18 27
Jm 10/10/06
10/032, 881